The use of the slotted link communication architecture described above can present challenges for accommodating asynchronous data transmissions within a subnet 10. For example, consider a file transfer process between a network master (e.g., server 12) and a client 16 that takes place in an operating environment under the control of the Windows™ operating system (or one of its variants) produced by Microsoft corporation of Redmond, Washington. In such a transfer, the requesting entity (i.e., the network resource, say a personal computer, requesting the transfer of data) initiates the transfer by sending a Server Message Block (SMB) protocol command to open the file on the target platform (e.g., a server or another personal computer storing the requested material). The target device responds with another SMB command and supplies the requesting device with a pointer to the requested file. The requesting device, using this pointer, then reads a block of N-bytes from the designated file, specifying the block size using an offset from the pointer provided by the target device. In response, the target device reads N-bytes worth of data from the subject file and delivers this information to a transmission control protocol/internet protocol (TCP/IP) (assuming this is the transfer protocol being used) layer that handles communication between the devices. The TCP/IP layer fragments the N-bytes of data into smaller TCP/IP packets and begins transmitting the packets to the requesting device across the communication channel. As the requesting device receives the packets, it transmits acknowledgements back to the target device. After the last packet for the N-byte transfer has been received and acknowledged, the requesting device transmits another SMB request for the next block of M-bytes from the subject file. M and N may be the same or different, and this process continues until the file transfer is complete.

Please replace the paragraph beginning at line 3, page 13, with the following rewritten paragraph:

Described herein is a scheme for avoiding latencies in asynchronous communications within a wireless communication channel of a computer network. The present scheme is generally applicable to a variety of network environments, but finds especially useful application in a wireless computer network which is located in a home environment. Thus, the present scheme will be discussed with reference to the particular

aspects of a home environment. However, this discussion should in no way be seen to limit the applicability or use of the present invention in and to other network environments and the broader spirit and scope of the present invention is recited in the claims which follow this discussion.

a2

Please replace the paragraph beginning at line 5, page 14, with the following rewritten paragraph:

As explained in greater detail in the above-cited co-pending application, when a client device 16 joins a subnet, the client receives a Connection Agreements (CAG) package from the network master device (e.g., server 12). This package includes, among other things, information regarding the forward and backward bandwidth (e.g., the slots of the channel) to which the new client 16 is entitled. In addition, the maximum number of bytes the new client 16 can send/expect in each data packet is set for each type of packet (e.g., video data, audio data, etc.). The Connection Agreements package may also contain information regarding the total number of data frames that the new client 16 needs to wait (i.e., before transmitting its traffic) from the start of server's transmission and the identification of the preceding client (i.e., the client that owns the preceding reverse transmission slot). The client is also assigned a unique session identifier (CS-ID).

a3

Please replace the paragraph beginning at line 20, page 15, with the following rewritten paragraph:

In conjunction with the use of an AST 54, a client 16 may also incorporate an early trigger timer (ETT) 55 to ensure that packets are ready for transmission within the subnet when the client's transmission slot arrives. That is, an ETT 55 may be used to advance the internal construction of a packet or packets for transmission, with the goal being to have those packets assembled and ready for transmission when the client's transmission slot becomes available. Thus the ETT 55 (which may be implemented as a conventional timer) triggers the process of formation of packets and their error protection bits, etc., using a packet construction engine 56 (which in some cases may be a part of client 16 or in other

a4

cases may be a part of radio 14) to keep a few packets, at least, prepared before the actual start of transmission. For example, in one embodiment packets could be assembled and stored for transmission one network frame in advance of their actual transmission. Such preassembly is helpful in avoiding the idle times on the channel when the first few packets for a transmission sequence are being formed. Although collecting data for one network frame in advance of transmission may penalize the system with a one network frame latency at the first transmission slot, it is expected that this period can be reduced to a few milliseconds.

Please replace the paragraph beginning at line 5, page 17, with the following rewritten paragraph:

Each client device may program an associated CCA timer 57 (see Figure 3) to a predetermined value. The network master may specify this value at the time a master-client connection is established and it generally represents a period of time that must expire before a network client is permitted to assume that another client is not using the channel. Now, suppose a client device is 5th in line for transmission after the master device and suppose it detects a clear channel (e.g., because its CCA timer 57 times out) after the device that is second in line has completed its transmission. Because the device is 5th in line, it cannot immediately begin its own transmissions (after all, there are still two client devices in line ahead of it). However, the 5th device may increment an associated CCA counter 58 at this time. If then both the 3rd and 4th devices are silent (i.e., if the 5th device's CCA timer 57 times out two more times in succession), then the 5th device will have again twice incremented its CCA counter 58 and may immediately start its own transmission on the 3rd CCA detection.

Please replace the paragraph beginning at line 4, page 19, with the following rewritten paragraph:

In order to provide a somewhat fair allocation scheme, in one embodiment each client is permitted to transmit only one packet in the idle time at the end of a network frame

52. This allows the devices in subnet 10 to take turns transmitting asynchronous (e.g., low priority) data over the channel. This protocol may be abandoned and a transmission commenced if a packet from a previous client is detected in the idle time and there is sufficient time in the idle period for a packet transmission. However, before any transmission in the idle time occurs, a device wishing to send data should allow sufficient time for the Q slot before the commencement of the next network frame.

Please replace the paragraph beginning at line 14, page 19, with the following rewritten paragraph:

If a device wishes to transmit multiple packets within the idle time of a single network frame, a two-step process may be invoked. First, the device transmits an initial packet in its regular space in the idle time, determined according to the above protocol. Then, the device may reprogram its T_{idle} time such that $T_{idle} = T_{CCA} * N$, where N is the total number of devices (including the master) in the subnet 10. The next idle time transmission for the subject device can then occur at this new T_{idle} time, provided sufficient time remains before the Q slot. If a packet from another device is received before this new T_{idle} time expires, the CCA may be reprogrammed to the time between the device's transmission and the reception of the newly received packet. This helps assure an equal opportunity for all devices in the subnet 10 to use the idle time for transmission of asynchronous data.

Please replace the paragraph beginning at lines 10 and 12, page 20, with the following rewritten paragraph:

Packet size will play a role in determining how many devices are afforded the opportunity to transmit within an idle time. That is, even if a device's time for transmission in an idle time has arrived, that device should not transmit if it detects an ongoing transmission of another device. Also, devices sending smaller packets during an idle time may be benefited to a lesser extent than those transmitting larger packets, as a device is only (usually) able to transmit one packet per idle time. To balance this equation, and